УДК 004.75

Системы пилотных заданий для интеграции Курчатовского суперкомпьютера в гетерогенную среду вычислительных ресурсов*

Д.Д. Дрижук[1], А.А. Пойда[1], Д.А. Олейник[2,3]

[1]Национальный Исследовательский Центр «Курчатовский Институт», Москва, Россия

[2]Университет Техаса в Арлингтоне, Техас, США

[3]Объединенный институт ядерных исследований, Дубна, Россия

Введение

Возрастающие объемы научных расчетов предъявляют все более высокие требования к используемым вычислительным мощностям. В качестве таковых продолжают использоваться грид-инфраструктуры, платформы облачных вычислений, суперкомпьютеры, университетские и исследовательские кластеры и т.д. Однако все чаще появляются задачи, для которых необходима интеграция и федерализация различных вычислительных ресурсов в единую гетерогенную вычислительную среду. Ярким примером такой задачи является задача по обработке, анализу и моделированию данных для экспериментов на Большом Адронном Коллайдере (БАК) [1].

В 2014 году в НИЦ «Курчатовский институт» была начата пионерская работа по созданию полномасштабной системы управления данными и заданиями в среде федеративных гетерогенных ресурсов. В качестве программной основы для системы управления был выбран подход, используемый в системе управления заданиями PanDA (“Production and Distributed Analysis”) [2], с 2005 года успешно используемой для обработки, анализа и моделирования данных эксперимента ATLAS [3], проводимого на БАК [1]. Одной из целей проекта является интеграция в разрабатываемую систему вычислительных ресурсов с различной архитектурой, включая суперкомпьютеры, платформы облачных вычислений, высокопроизводительные вычислительные кластеры и т.п., а также адаптация разрабатываемой системы к различным областям наук, требующим высокоинтенсивных вычислений, включая физику высоких энергий и не только.

Одним из ключевых программных компонентов, используемым в PanDA для интеграции гетерогенных вычислительных ресурсов, является система пилотных заданий, являющаяся медиатором между архитектурно-зависимыми вычислительными агентами и платформо-независимым ядром системы [4]. Чтобы интегрировать Курчатовский суперкомпьютер в систему, потребовалось расширить базовую схему пилотных заданий PanDA и адаптировать ее к архитектуре суперкомпьютера.

В данной статье описана модификация и адаптация базовой схемы пилотных заданий PanDA для интеграции суперкомпьютера НИЦ «Курчатовский институт» в разрабатываемую систему управления данными и заданиями в среде федеративных гетерогенных ресурсов.

Подсистема пилотных заданий PanDA

Одним из ключевых компонентов системы PanDA является подсистема пилотных заданий, реализующих операцию поздней привязки. Пилотные задания запускаются на рабочих узлах вычислительной системы (например, вычислительного кластера) перед тем, как туда будет отправлена пользовательская задача. После запуска пилотные задания собирают информацию о каждом конкретном вычислительном ресурсе и проверяют его на предмет соответствия минимально-необходимым требованиям. Если проверка успешна, то пилотное задание запрашивает у сервера пользовательскую задачу, получает ее, запускает на вычислительных ресурсах, контролирует процесс выполнения и после ее окончания

*□ От лица исполнителей проекта BigPanDA

передает серверу результаты. Таким образом, задача начинает работу в уже проверенной и, при необходимости, подготовленной среде, привязываясь к ресурсу на максимально близкой к запуску стадии. 'Поздняя привязка' рабочих задач к месту вычислений предотвращает задержки и отказы, и максимизирует гибкость выделения ресурсов задачам при помощи динамического состояния обрабатывающих ресурсов и приоритетов задач. Пилотные задания являются также основным 'изолирующим слоем' системы, инкапсулирующим сложные неоднородные среды и интерфейсы, с которыми взаимодействуют слои более высокого уровня.

На рисунке 1 приведена базовая схема работы пилотного задания PanDA.

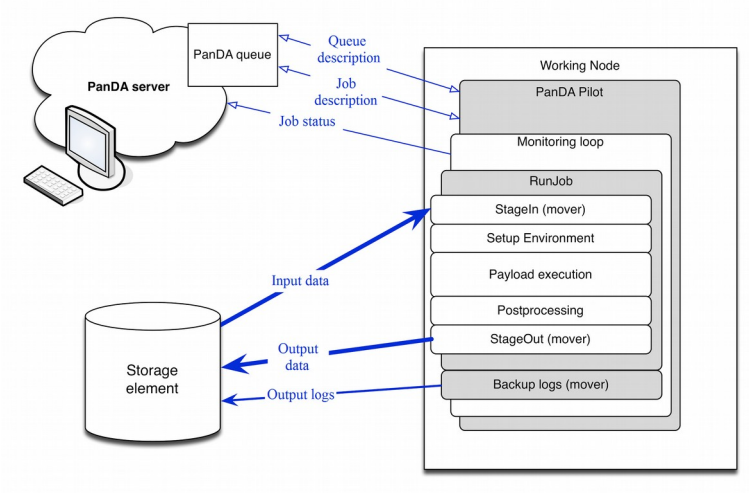

Рисунок 1. Базовая схема работы пилотных заданий PanDA

Схема, представленная на рисунке 1, является классическим вариантом, используемым в системе PanDA для выполнения задач эксперимента ATLAS. Основными компонентами модуля пилотного задания являются:

- Mover, осуществляющий загрузку в локальное хранилище входных данных, необходимых для выполнения пользовательской задачи, а также выгрузку и регистрацию результирующих данных.
- RunJob, запрашивающий и получающий пользовательскую задачу с сервера и запускающий ее на выполнение на ресурсах вычислительной среды.
- Monitor, предназначенный для контроля выполнения задачи и работой всех компонент пилотного задания.

Схема работы пилотного задания, представленная на рисунке 1, предполагает, что вычислительная среда, в которой будет выполняться пользовательская задача, имеет доступ к внешней сети (что позволяет напрямую обращаться к серверу и хранилищам данных). Такая конфигурация характерна для отдельных машин, кластеров, инфраструктур облачных вычислений. В этом случае пилотное задание запускается прямо на рабочих узлах вычислительной среды и работает с процессами и данными локально: данные загружаются в локальную файловую систему, обрабатываются как локальные файлы, пользовательская задача запускается локально. Отрицательным моментом схемы является сложность запуска параллельных задач, использующих, например, интерфейс передачи сообщений MPI, так как пользовательская задача запускается на ресурсах, уже выделенных пилотному заданию. Когда пилотное задание резервирует ресурсы, информация о том, какая пользовательская задача будет делегирована данному пилотному заданию и сколько узлов ей требуется, еще недоступна, так как используется позднее связывание.

Расширение базовой схемы пилотных заданий PanDA для интеграции суперкомпьютера НИЦ «Курчатовский институт»

Чтобы упростить запуск параллельных многоядерных задач, а также отказаться от требования доступности внешней сети с узлов вычислительной среды авторы расширили базовую схему PanDA «удаленным» режимом работы. Схема работы пилотных заданий в удаленном режиме представлена на рисунке 2

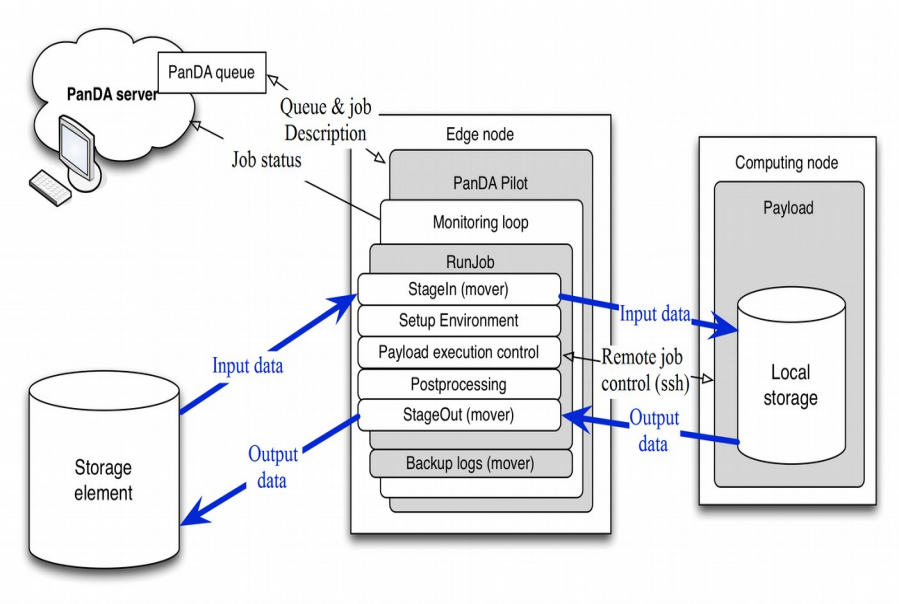

Рисунок 2. Схема работы пилотных заданий на удаленном узле (изолированная вычислительная среда)

Схема, представленная на рисунке 2, предназначена для работы с вычислительными средами, в которых рабочие узлы не имеют доступа ко внешним ресурсам. Такая конфигурация характерна для большинства суперкомпьютеров. В этом случае для запуска пилотного задания требуется промежуточный узел, выступающий в роли шлюза между сервером и вычислительной средой. Проверка наличия ресурсов и их доступность проверяются до получения пользовательской задачи, но захват ресурсов для нее осуществляется после ее получения. В результате, пилотное задание может зарезервировать требуемый объем ресурсов при запуске пользовательской задачи, что позволяет запускать параллельные задачи. Входные и выходные файлы пересылаются через узел пилотного задания, на котором буферизуются. Отрицательными моментами данной схемы является резервирование ресурсов только после получения задачи от сервера (что увеличивает вероятность сбоя), а также повышенная нагрузка на узел пилотных заданий, для которого требуется большой объем ресурсов, в частности свободного места на жестком диске.

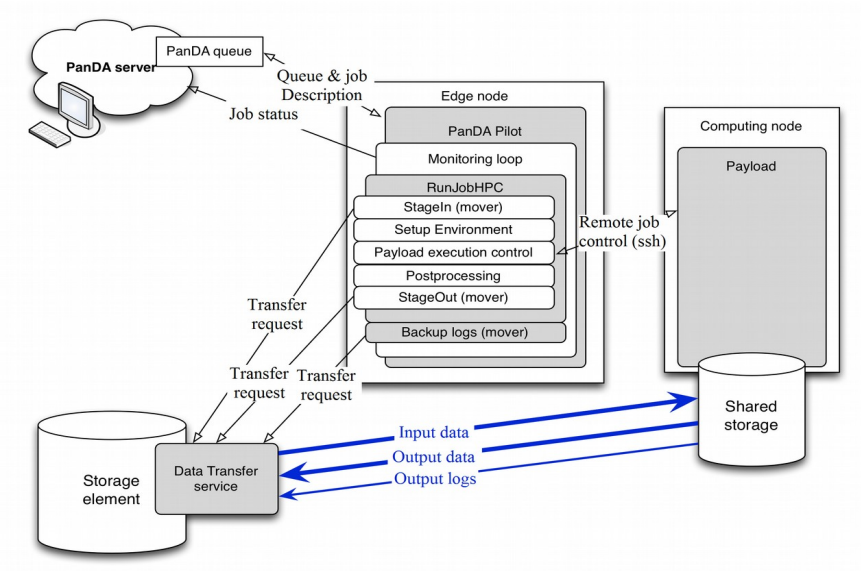

Рисунок 3. Схема работы пилотных заданий на удаленном узле (вычислительная среда прямой доступ ко внешнему хранилищу)

Схема, представленная на рисунке 3, является промежуточным вариантом между базовой схемой и схемой «удаленного» режима. В данном случае вычислительная среда так же, как и в базовой схеме, имеет доступ ко внешним источникам, либо к широкополосному каналу связи с общим хранилищем. Это дает возможность перемещать входные данные и результаты вычислений между хранилищами и вычислительной средой напрямую, что практически устраняет нагрузку на узел пилотных заданий. При этом сохраняется возможность запуска параллельных задач за счет того, что пилотное задание запускается на удаленном узле.

Первая из трех приведенных схем (Рисунок 1) является основной схемой в системе PanDA. Вторая схема (Рисунок 2) в несколько модифицированном частном виде также используется в PanDA для подключения изолированных суперкомпьютеров, где идёт резервирование блока задач под имеющиеся ресурсы. Последняя схема (Рисунок 3) - оригинальная разработка, которая позволяет использовать широкополосную связь с файловым хранилищем, как и позднее связывание.

Интеграция суперкомпьютера НИЦ «Курчатовский институт»

Суперкомпьютер НИЦ «Курчатовский институт» был интегрирован в экспериментальную версию системы управления данными и заданиями в среде федеративных гетерогенных ресурсов.

Суперкомпьютер НИЦ “Курчатовский институт – высокопроизводительный вычислительный кластер второго поколения с пиковой производительностью 122,9 TFLOPS сдан в эксплуатацию с сентября 2011 года. Кластер состоит из 1280 счётных двухпроцессорных узлов, объединенных высокопроизводительной сетью передачи данных и сообщений InfiniBand DDR, имеет суммарную оперативную память 20,5 Тбайт и систему хранения данных на 144 Тбайт. На счётных узлах кластера установлена операционная система Linux (CentOS). Система хранения данных построена на параллельной файловой системе Lustre 2.0. Для управления распределением ресурсов и выполнением счетных заданий используется менеджер ресурсов SLURM.

Так как суперкомпьютер НИЦ “Курчатовский институт” имеет доступ во внешние сети, то для его интеграции мы использовали схему, представленную на рисунке 3. Более детальная схема интеграции представлена на рисунке 4.

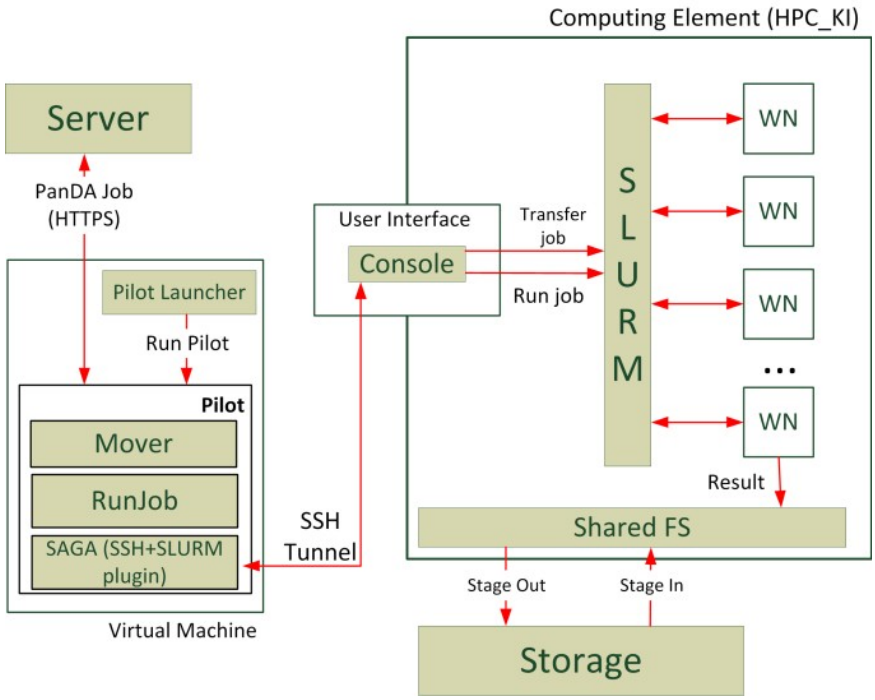

Рисунок 4. Детальная схема интеграции суперкомпьтера НИЦ “Курчатовский институт” в режиме «один пилот на много ядер»

Пилотные задания запускаются на виртуальной машине, с которой доступ на суперкомпьютер осуществляется по протоколу SSH через машину пользовательского интерфейса, с которой можно ставить задачи в очередь SLURM. Для загрузки входных данных из удаленного хранилища и выгрузки результатов обработки используются отдельные задачи.

В то же время мы использовали базовую схему работы пилотных заданий (Рисунок 1) для работы с сервером PanDA, установленным в CERN, и для запуска задач эксперимента ATLAS. Более детальная схема подключения суперкомпьютера НИЦ “Курчатовский институт” для выполнения задач эксперимента ATLAS представлена на рисунке 5.

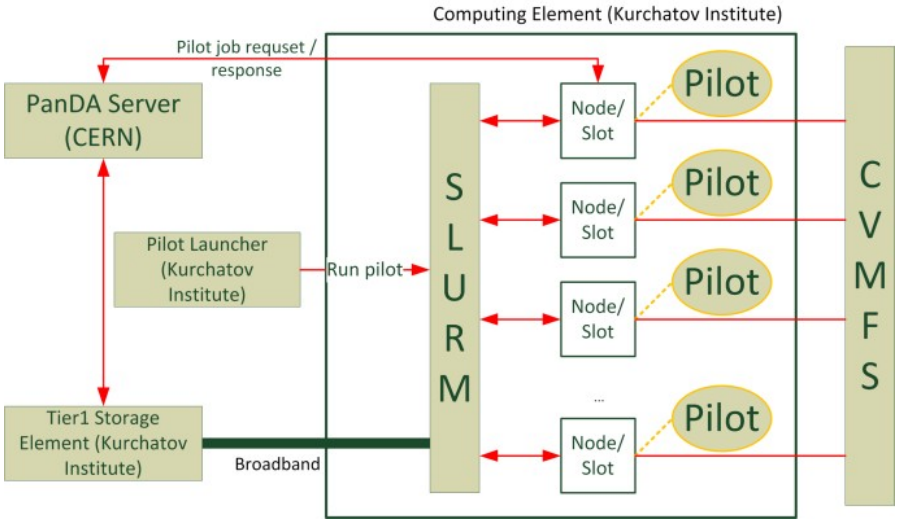

Рисунок 5. Схема интеграции суперкомпьютера НИЦ «Курчатовский институт» для выполнения задач эксперимента ATLAS в режиме «один пилот на одно ядро»

Pilot Launcher запускает пилотные задания последовательно из расчета – одно задание на одно ядро. Для того чтобы выполнялись пользовательские задания по обработке данных эксперимента ATLAS, требуется программное обеспечение, доступное на подключаемой удаленной файловой системе CVMFS. Входные и выходные данные загружаются непосредственно в Grid Storage Element Tier1 НИЦ “Курчатовский институт”.

Для тестирования разработанных схем интеграции было использовано два типа задач: однопроцессорные ATLAS задачи и многоядерные модельные задачи в области биоинформации. В качестве модельных задач были выбраны задачи выравнивания и сборки генома с помощью программного обеспечения Bowtie2 [5] и Abyss [6]. Непосредственные вычисления осуществляются на суперкомпьютере НИЦ “Курчатовский институт”. Первые тесты показали успешные результаты. С помощью пакета Quast [7] удалось расширить возможности мониторинга по пост-обработке выходных файлов - были сгенерированы специальные страницы отчетов успешности выполнения задач.

Заключение

В результате работы была модифицирована и расширена базовая схема пилотных заданий PanDA. Разработанные схемы были адаптированы для интеграции суперкомпьютера НИЦ «Курчатовский институт» в систему управления данными и заданиями в среде федеративных гетерогенных ресурсов.

Разработанные схемы позволяют запускать многопроцессорные задачи, которые существенно расширяют класс обрабатываемых задач и могут быть использованы в биологии для задач обработки данных геномного секвенирования, в астрофизике для изучения состава космических лучей, поиска антиматерии и тёмной материи, и т.п.

Разработанные схемы достаточно абстрактны и могут быть использованы для интеграции других вычислительных ресурсов, включая суперкомпьютеры и платформы облачных вычислений.

Благодарности

Данная работа выполнена в рамках мега-гранта правительства РФ, контракт No 14.Z50.31.0024. Результаты работы были получены с использованием вычислительных ресурсов МВК НИЦ «Курчатовский институт» (http://computing.kiae.ru/).

Литература

[1] Worldwide LHC Computing Grid. Сайт проекта. Электронный ресурс. URL: http://wlcg.web.cern.ch/

[2] Maeno, Tadashi. "PanDA: distributed production and distributed analysis system for ATLAS." Journal of Physics: Conference Series. Vol. 119. No. 6. IOP Publishing, 2008.

[3] Collaboration, A. T. L. A. S., and G. Aad. "The ATLAS experiment at the CERN large hadron collider." J. Instrum 3 (2008): S08003.

[4] T. Maeno et al. “Evolution of the ATLAS PanDA workload management system for exascale computational science” 2014 J. Phys.: Conf. Ser. 513 032062.

[5] Программа для выравнивания последовательностей. Сайт проекта. Электронный ресурс. URL: http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml

[6] Simpson, Jared T., et al. "ABySS: a parallel assembler for short read sequence data." Genome research 19.6 (2009): 1117-1123.

[7] Программа для оценки качества сборок. Сайт проекта. Электронный ресурс. URL: http://bioinf.spbau.ru/ru/quast